

## REMARKS

In the Office Action mailed July 31, 2008, claims 1 and 2 were objected to due to informalities; claims 1-8 and 14-21 were rejected under §112; claims 22-29 were rejected under §101; and claims 1-29 were rejected under §103 as being unpatentable over Milillo in view of Asselin and Taylor.

In this Response, claim 1 was amended to address the first informality and the other independent claims 9, 14 and 22 were amended in the same manner. Claim 2 was amended to address the second informality and claim 23 was amended in the same manner. Claim 2 was also amended to address the third informality.

Claim 22 has been amended to address the §101 rejection by specifying that the recited computer readable medium is limited to a “storage” medium.

The Applicant respectfully again traverses the rejection under §103. The Applicant again asserts that the FlashCopy function to which the present invention pertains is different from the concurrent copy discussed in some of the cited references, such as Asselin. The concurrent copy function enables a copy of data to be made while an application is updating that data. The copy is “in a consistent form, as it existed before the updates took place ... as though the updates had not occurred.” (Asselin, pg. 1) Data is copied concurrently with normal operations. By contrast, when the FlashCopy function used in the present invention is invoked, a FlashCopy relationship is established between a source volume (to which a copy of the data has been transferred from the primary storage unit, as explained in paragraph 23 of the Specification) and a target volume, thereby providing a mapping of the source volume and target volume to allow a point-in-time copy (based on consistency groups) of the source volume to be copied to the target volume. The operation is nearly instantaneous and copies are “immediately” available for read and write access. Consequently, the Applicant reiterates that FlashCopy and concurrent copy are different functions and the latter should not be used to render the former obvious.

The Office Action also asserts that Asselin teaches “imposing a write-inhibit indicator on each FlashCopy source volume” (page 10). Page 3 of Asselin is cited: “The system serializes access the data being dumped or copied just long enough for the

concurrent session to be initialized.” However, the term “serialize” implies that access is continues, though perhaps in a different fashion than during a normal operation and thus teaches away from write-inhibiting.

With respect to the assertion that Taylor teaches committing a FlashCopy operation if the preparation is successful and reverting the FlashCopy operation is the preparation is not, Taylor teaches merely “releasing the database from backup preparation” (paragraph 40). “Releasing” a database is different from the claimed “reverting” the FlashCopy operation. An issue which the claimed invention addresses is that:

Due to the distributed nature of the environment [in which a consistency group may be distributed over many storage volumes in many storage controllers], FlashCopy commands to the source volumes (on the secondary unit 104) do not execute simultaneously. Consequently, once a FlashCopy operation begins, the FlashCopy target volumes 120 become inconsistent until the FlashCopy of all source volumes 118 is completed: some source/target volume pairs may have completed the FlashCopy, others may be in process of executing the FlashCopy command and still other may not have received the command yet. (Specification, paragraph 26).

If a timeout, warmstart or other such event occurs during the above-described transition phase, neither the source volumes nor the target volumes are consistent and reversion to the prior consistency group is no longer possible. Thus, the data is at risk of loss. As described in paragraph 31, when the claimed invention is implemented, if the FlashCopy preparation of any target volume is not successful, no change is made with respect to those volumes. However, the target volumes which were successfully prepared are reverted to their previous state. Therefore, “the FlashCopy target volumes ... retain the prior consistency group in uncorrupted form” and there is no risk of loss of data.

Dependent claims recite, and paragraph 32 describes, that the success or failure of the preparation of each target volume may be determined after the attempt is made

to prepare each volume or may be determined after attempts have been made to prepare all of the volumes.

Neither Taylor nor either of the other references address the same problem as the claimed invention. None of the references disclose making a determination of whether any of the target volumes has been successfully prepared for a FlashCopy operation. And, none of the references disclose reverting any successfully prepared target volumes to their prior state in the event that the preparation of any volume is unsuccessful. Consequently, the cited combination of references does not render the claims obvious.

As substantially the same grounds for rejection were asserted against all of the independent claims, the foregoing comments apply equally to those claims. In addition, the Applicant respectfully asserts that the dependent claims are further allowable based on the allowability of the respective independent claims.

For the foregoing reasons, the claims are believed to be allowable, the Application is believed to be in condition for allowance and a favorable Office Action is requested. The Examiner is encouraged to contact the undersigned by telephone if a conversation would expedite prosecution of this case.

This constitutes a request for any needed extension of time. No fee is believed to be due in this instance. The undersigned hereby authorizes the charge of any deficiency of fees submitted herewith, or the credit of any overpayment, to deposit account number 09-0449.

Respectfully Submitted,

/Dan Shifrin/  
Dan Shifrin, #34,473  
Law Office of Dan Shifrin  
14081 West 59<sup>th</sup> Ave  
Arvada, Colorado 80004  
303-403-4510  
303-785-8795 (Fax)

cc: IBM- Tucson